九州工業大学学術機関リポジトリ

Kyushu Institute of Technology Academic Repository

Title	Worst and best irredundant sum-of-products expressions
Author(s)	Sasao, Tsutomu; Butler, J.T.
Issue Date	2001-09
URL	http://hdl.handle.net/10228/615
Rights	"©2001 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE."

Kyushu Institute of Technology Academic Repository

Worst and Best Irredundant Sum-of-Products Expressions

Tsutomu Sasao, *Fellow*, *IEEE*, and Jon T. Butler, *Fellow*, *IEEE*

Abstract—In an irredundant sum-of-products expression (ISOP), each product is a prime implicant (PI) and no product can be deleted without changing the function. Among the ISOPs for some function f, a worst ISOP (WSOP) is an ISOP with the largest number of PIs and a minimum ISOP (MSOP) is one with the smallest number. We show a class of functions for which the Minato-Morreale ISOP algorithm produces WSOPs. Since the ratio of the size of the WSOP to the size of the MSOP is arbitrarily large when n, the number of variables, is unbounded, the Minato-Morreale algorithm can produce results that are very far from minimum. We present a class of multiple-output functions whose WSOP size is also much larger than its MSOP size. For a set of benchmark functions, we show the distribution of ISOPs to the number of PIs. Among this set are functions where the MSOPs have almost as many PIs as do the WSOPs. These functions are known to be easy to minimize. Also, there are benchmark functions where the fraction of ISOPs that are MSOPs is small and MSOPs have many fewer PIs than the WSOPs. Such functions are known to be hard to minimize. For one class of functions, we show that the fraction of ISOPs that are MSOPs approaches 0 as n approaches infinity, suggesting that such functions are hard to minimize.

Index Terms—Logic minimization, complete sum-of-products expressions, irredundant sum-of-products, multiple-output functions, heuristic minimization, prime implicants, symmetric functions, minimum sum-of-products expressions, worst sum-of-products expressions, graph enumeration, minimally strongly connected digraphs.

1 INTRODUCTION

TWO-LEVEL logic minimization is a basic problem in logic synthesis. Although algorithms exist that obtain the exact minimum sum-of-products expressions (MSOP) for a large set of functions [7], [8], the majority of practical systems use heuristic logic minimization algorithms. These produce irredundant sum-of-products expressions (ISOPs) that are not necessarily minimum. For example, PRESTO [4], [33], MINI [15], ESPRESSO [3], and others [10], [24] produce nonminimum ISOPs.

An ISOP is the OR of prime implicants (PIs) such that deleting any PI changes the function. For example, two expressions $x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee \bar{x}_1 x_3$ and $x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee \bar{x}_1 x_2 \vee \bar{x}_1 x_3$ are both ISOPs for the same function (see Fig. 1a and Fig. 1b). The first is an MSOP and the second is a worst ISOP (WSOP), an ISOP with the largest number of PIs. Most practical logic synthesis algorithms generate ISOPs at some point, so an understanding of ISOPs is crucial.

In this paper, we show classes of functions, where the ratio of the WSOP size (number of PIs) to the MSOP size is arbitrarily large when the number of variables is unbounded. We show that the Minato-Morreale algorithm [19], [20] produces WSOPs for this class. We also show an n-variable multiple-output function whose MSOP size is at most $2n$ and whose WSOP size is at least 2^n.

- T. Sasao is with the Department of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka 820, Japan. E-mail: sasao@cse.kyutech.ac.jp.
- J.T. Butler is with the Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA 93943-5121. E-mail: butler@cs.nps.navy.mil.

Manuscript received 14 June 1999; accepted 22 Mar. 2001.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 110047.

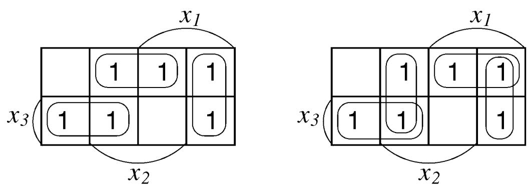
We also show an algorithm that produces *all* ISOPs for a given function. When applied to benchmark functions, we notice a correlation between the degree of difficulty in determining an MSOP and the distribution of ISOPs to the number of PIs. For example, easily minimized functions tend to have a larger proportion of ISOPs that are MSOPs.

We also show an analysis of a class of functions with respect to the number of MSOPs and WSOPs. For this class, the number of ISOPs that are MSOPs and WSOPs is $n(n-1)!$ and n^{n-2}, respectively. Since there are many more WSOPs than MSOPs, it suggests that such functions are difficult to minimize.

This paper is organized as follows: Section 2 shows definitions and basic properties. Section 3 considers the MSOPs and WSOPs of specific functions. The ratio of the number of PIs in a WSOP to that of the MSOP is arbitrarily large when n is unbounded for these functions. Section 4 presents a class of multiple-output functions for which there is a large disparity between the number of PIs in an MSOP and in a WSOP. Section 5 focuses on the distribution of the number of ISOPs to the number of PIs required. It presents a method to derive all ISOPs of a given function. Section 6 shows experimental results obtained from this method. Section 7 shows how this distribution can be obtained analytically for specific functions. Section 8 concludes the paper.

2 DEFINITIONS AND FUNDAMENTAL RESULTS

In the discussions to follow, we will often use symmetric functions.



(a) (b)

Fig. 1. Karnaugh maps for $ST(3,1)$. (a) MSOP. (b) WSOP.

Definition 2.1. *S_A^n, a (**totally**) **symmetric function**, is 1 if m of its n variables are 1, where $m \in A \subseteq \{1, 2, \ldots, n\}$ and is 0 otherwise.*

Example 2.1. The AND and OR of n variables are symmetric functions, represented by $S_{\{n\}}^n$ and $S_{\{1,2,\ldots n\}}^n$, respectively.

Definition 2.2. *x and $\bar{x}$ are literals of a variable x. The AND of literals is a **product**. The OR of products is a **sum-of-products expression** (SOP).*

Definition 2.3. *A **prime implicant** (**PI**) of a function f is a product that implies f such that the deletion of any literal from the product results in a new product that does not imply f.*

Definition 2.4. *A **complete sum-of-products expression** (**CSOP**) [2], [22] of a function f is the SOP of all PIs of f.*

Definition 2.5. *An **irredundant sum-of-products expression** (**ISOP**) is an SOP where each product is a PI and no PI can be deleted without changing the function represented by the expression.*

Definition 2.6. *Among the ISOPs for f, the one with the largest number of PIs is a **worst ISOP** (**WSOP**) and the one with the smallest number of PIs is a **minimum SOP** (**MSOP**).*

Defintion 2.7. *The **size** of an SOP is the number of PIs in the SOP. The size of a CSOP, WSOP, and MSOP of function f is denoted as $\tau(CSOP : f)$, $\tau(WSOP : f)$, and $\tau(MSOP : f)$, respectively.*

The following is well known.

Theorem 2.1 [13], [22]. *For any switching function of n variables, $\tau(MSOP : f) \leq 2^{n-1}$.*

This upper bound is firm. For example, the exclusive OR function, $f_{EXOR} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$, has 2^{n-1} minterms, all of which are PIs, and, so, $\tau(MSOP : f_{EXOR}) = 2^{n-1}$.

Further, $\tau(WSOP : f_{EXOR}) = 2^{n-1}$, there being only one ISOP. It is tempting to believe that $\tau(WSOP : f) \leq 2^{n-1}$, for *any* f. Indeed, Meo [18] conjectured this in the mid 1960s. However, a counterexample was published in Russian in 1962 by Yablonski [37] (which was reported in English by Kautz [16] in 1966). Specifically, Yablonski showed:

Theorem 2.2 [37]. *There exists a switching function on n variables where $\tau(WSOP : f) > 2^{n-1}$.*

by showing an ISOP for $S_{\{0,1,3,4,6,7\}}^7$ with 70 PIs. This is six more than the upper bound of $2^{7-1} = 64$ as conjectured by Meo. As we have not seen a proof of this in English, we include one here. We extend Yablonski's result by showing that his ISOP is a WSOP.

Lemma 2.1.

$$\tau(WSOP : S_{\{0,1,3,4,6,7\}}^7) = 70.$$

Proof. See the Appendix. □

3 WSOPs and MSOPs for Specific Functions

Definition 3.1. *Let $ST(n,k)$ be a symmetric function of n-variables $x_1, x_2, \ldots, x_n$ such that*

$$ST(n,k) = \begin{cases} 1 & k \leq \sum_{i=1}^{n} x_i \leq n-k \\ 0 & otherwise, \end{cases}$$

where $\sum_{i=1}^{n} x_i$ is the number of variables that are 1 and $n \geq 2k$.

Example 3.1. $ST(n,0) = 1$. $ST(n, \frac{n}{2})$, for even n, is the OR of all minterms with exactly half of the variables complemented.

Lemma 3.1. *$ST(n,k)$ can be represented as*

$$ST(n,k) = S_{\{k,k+1,\ldots,n\}}^n S_{\{0,1,\ldots,n-k\}}^n.$$

Example 3.2.

$$\begin{aligned} ST(n,1) &= S_{\{1,2,\ldots,n\}}^n S_{\{0,1,\ldots,n-1\}}^n \\ &= (x_1 \vee x_2 \vee \cdots \vee x_n)(\bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_n) \end{aligned}$$

and

$$\begin{aligned} ST(n,2) &= S_{\{2,3,\ldots,n\}}^n S_{\{0,1,\ldots,n-2\}}^n \\ &= (x_1x_2 \vee x_1x_3 \vee \cdots \vee x_{n-1}x_n) \\ &\quad (\bar{x}_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3 \vee \cdots \vee \bar{x}_{n-1}\bar{x}_n). \end{aligned}$$

We are interested in the sizes of the CSOP, an MSOP, and a WSOP for $ST(n,k)$. Voight and Wegener [36] consider the CSOP and MSOP sizes for general symmetric functions, stating expressions, and outlining a proof. Our next result gives the CSOP, MSOP, and WSOP sizes for $ST(n,k)$ functions. A complete proof is given in the Appendix.

Theorem 3.1.

1. $\tau(CSOP : ST(n,k)) = \binom{n}{k,\, n-2k,\, k} = \frac{n!}{k!(n-2k)!k!}$.
2. $\tau(MSOP : ST(n,k)) = \binom{n}{k}$.
3. $\tau(WSOP : ST(n,k)) \geq 2\binom{n}{k} - \binom{2k}{k}$.

Proof. See the Appendix. □

In the proof of Lemma 2.1, we showed that $\tau(WSOP : ST(7,3)) = 56$, which is six more than the lower bound given in 3 of Theorem 3.1. Therefore, for $n = 7$ and $k = 3$, the lower bound is not tight. However, for $k = 1$, the lower bound is exact. That is,

Theorem 3.2.

$$\tau(WSOP : ST(n,1)) = 2n - 2.$$

Proof. See the proof of Theorem 7.3 in the Appendix. □

A special case of these theorems occurs when $n = 3$ and $k = 1$.

Example 3.3. $ST(3,1) = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$ has the following properties:

1. $\tau(CSOP : ST(3,1)) = 6$.
2. $\tau(MSOP : ST(3,1)) = 3$.
3. $\tau(WSOP : ST(3,1)) = 4$.

Fig. 1a and Fig. 1b show the MSOP and WSOP of $ST(3,1)$, respectively. Interestingly, the ISOP generator of Minato [19], which is based on Morreale's [20] algorithm produces a WSOP for $ST(3,1)$ instead of an MSOP. This will be discussed in more detail later.

Definition 3.2. *The* ***redundancy ratio*** *of a function f is*

$$\rho(f) = \frac{\tau(WSOP : f)}{\tau(MSOP : f)}.$$

The ***normalized redundancy ratio*** *of an n-variable function f is*

$$\sigma(f) = \sqrt[n]{\rho(f)},$$

where $\tau(WSOP : f)$ and $\tau(MSOP : f)$ are the sizes of WSOPs and MSOPs. If this ratio is small, any *logic minimization algorithm will do well since, even if a WSOP is generated, it is not much worse than an MSOP. On the other hand, a large ratio suggests that care should be exercised. The normalized redundancy ratio is normalized with respect to the number of variables. It is a convenience; it allows one to compare the redundancy ratio of two functions with a different number of variables.*

From the expressions for $\tau(MSOP : ST(n,k))$ and $\tau(WSOP : ST(n,k))$ given in Theorem 3.1, we can state:

Theorem 3.3.

$$\rho(ST(n,k)) \geq 2 - \frac{\binom{2k}{k}}{\binom{n}{k}},$$

$$\sigma(ST(n,k)) \geq \sqrt[n]{2 - \frac{\binom{2k}{k}}{\binom{n}{k}}}.$$

From the expressions for $\tau(MSOP : ST(n,1))$ and $\tau(WSOP : ST(n,1))$ given in Theorems 3.1 and 3.2, respectively, we can state:

Theorem 3.4.

$$\rho(ST(n,1)) = 2 - \frac{2}{n},$$

$$\sigma(ST(n,1)) = \sqrt[n]{2 - \frac{2}{n}}.$$

Table 1 shows the values of ρ and σ for $ST(n,1)$, where $2 \leq n \leq 8$. It can be seen that σ takes its maximum value when $n = 4$. That is, as n increases above 2, σ first increases, peaking at 4, and then it continually decreases.

From Theorem 3.4, ρ is monotone increasing with an upper limit of 2. Thus, for $ST(n,1)$ functions, the number of PIs in a WSOP is never more than two times the number of PIs in an MSOP. An important question is whether there exist functions where ρ is larger than 2. Indeed, we show a class of functions in which ρ increases without bound as n increases. This has important consequences for heuristics that produce ISOPs. For such heuristics there is the prospect of generating an ISOP whose size is much larger than the minimum. We consider this topic now.

TABLE 1
ρ and σ for $ST(n,1)$ versus n

n	ρ	σ
2	1.0000	1.0000
3	1.3333	1.1006
4	1.5000	1.1066
5	1.6000	1.0986
6	1.6667	1.0889
7	1.7143	1.0800
8	1.7500	1.0725
$\rightarrow \infty$	$\rightarrow 2$	$\rightarrow 1$

Definition 3.3. *Let $ST(m,k)^r$ be the $n = m \cdot r$-variable function*

$$ST(m,k)^r(x_1, x_2, \ldots, x_{mr}) = \bigwedge_{i=1}^{r} ST(m,k)(x_{m(i-1)+1}, x_{m(i-1)+2}, \ldots, x_{mi}),$$

where $\bigwedge_{i=1}^{r}$ is the AND (product) of r functions.

Theorem 3.5. *$ST(m,k)^r$ has the following properties:*

1. $\tau(CSOP : ST(m,k)^r) = \binom{m}{k, m-2k, k}^r = (\frac{m!}{k!(m-2k)!k!})^r$.
2. $\tau(MSOP : ST(m,k)^r) = \binom{m}{k}^r$.
3. $\tau(WSOP : ST(m,k)^r) \geq [2\binom{m}{k} - \binom{2k}{k}]^r$.

Proof. See the Appendix. □

For $k = 1$, we have:

Theorem 3.6.

$$\tau(WSOP : ST(m,1)^r) = 2^r(m-1)^r.$$

Example 3.4. For $m = 3$ and $k = 1$, $ST(3,1)^r$ has 6^r PIs, $\tau(MSOP : ST(3,1)^r) = 3^r$, and

$$\tau(WSOP : ST(3,1)^r) = 4^r.$$

We have:

Theorem 3.7.

$$\rho(ST(m,k)^r) \geq \left[2 - \frac{\binom{2k}{k}}{\binom{m}{k}}\right]^r,$$

$$\rho(ST(m,1)^r) = 2^r\left(1 - \frac{1}{m}\right)^r.$$

Example 3.5. For $m = 4$ and $k = 1$, we have

$$\rho(ST(4,1)^r) = (1.5)^r.$$

From this, it can be seen that ρ becomes arbitrarily large as r approaches infinity. In this example, there are $n = 4r$

variables. This represents a class of functions for which ρ grows without bound as the number of variables grows.

4 Extension to Multiple-Output Functions

In the case of multiple-output functions, minimization of AND-OR two-level networks or programmable logic arrays (PLAs) can be done using characteristic functions [26], [27], [29].

Definition 4.1. *For an n-variable function with m output values,*

$$f_j(x_1, x_2, \ldots, x_n)\ (j = 0, 1, \ldots, m-1),$$

form an $(n+1)$-variable two-valued single output function $F(x_1, x_2, \ldots, x_n, X_{n+1})$, where x_i is a binary valued variable, for $1 \leq i \leq n$ and X_{n+1} takes m values such that $F(x_1, x_2, \ldots, x_n, j) = 1$ iff $f_j(x_1, x_2, \ldots, x_n) = 1$ $(j = 0, 1, \ldots, m-1)$. Then, F represents all and only the permitted combinations of inputs and nonzero output values of f. F is called the ***characteristic function*** *(for nonzero outputs).*

The significance of the characteristic function is seen in Theorem 4.1 below.

Definition 4.2. *X^S is a literal, where X takes a value in $\{0, 1, \ldots, p-1\}$ and $S \subseteq \{0, 1, \ldots, p-1\}$ such that $X^S = 1$ if $X = a \in S$ and $X^S = 0$, otherwise. A logical product of literals that contains at most one literal for each variable is a product term. Products combined with OR operators form a sum-of-products expression (SOP). A Prime implicant (PI), irredundant sum-of-products expression (ISOP), worst ISOP (WSOP), and minimum SOP (MSOP) are defined in a manner similar to the two-valued case.*

Theorem 4.1 [15], [27], [29]. *The number of AND gates in the minimum AND-OR two-level network for the function $(f_0, f_1, \ldots, f_{m-1})$ is equal to the number of PIs in the MSOP for the characteristic function F.*

Definition 4.3. *An n-bit decoder has n inputs $x_1, x_2, \ldots, x_n$, and 2^n outputs $f_0, f_1, \ldots, f_{2^n-1}$, where $f_i = 0$ iff the binary number representation of $x_1 x_2, \ldots, x_n$ is i.*

Example 4.1. The 4-bit decoder has 16 outputs, as follows:

$$\begin{aligned} f_0 &= x_1 \vee x_2 \vee x_3 \vee x_4, \\ f_1 &= x_1 \vee x_2 \vee x_3 \vee \bar{x}_4, \\ &\vdots \\ f_{15} &= \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4. \end{aligned}$$

Definition 4.4. *DEC_n is the characteristic function of an n-bit decoder.*

Example 4.2. DEC_4 is shown in positional cube notation in the upper table of Fig. 2. That is, each entry in this table is a prime implicant of DEC_4, where x_i appears as $\bar{x}_i$, x_i, or don't care (absent) if the corresponding entry is 10, 01, or 11, respectively. For X_5, the entry 0111111111111111 is the literal $X_5^{\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}}$, etc. Therefore, the first entry, 10 10 10 10 0111111111111111, corresponds to the prime implicant $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4X_5^{\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}}$. Collectively, the 16 entries in the upper table of Fig. 2 represent an ISOP of DEC_4 with 16 PIs. An ISOP for DEC_4 with only eight PIs exists, as shown in the lower table of Fig. 2.

ISOPs with 16 PIs

x_1	x_2	x_3	x_4	X_5
01	01	01	01	01 $\cdots$ 15
10	10	10	10	0111111111111111
10	10	10	01	1011111111111111
10	10	01	10	1101111111111111
10	10	01	01	1110111111111111
10	01	10	10	1111011111111111
10	01	10	01	1111101111111111
10	01	01	10	1111110111111111
10	01	01	01	1111111011111111
01	10	10	10	1111111101111111
01	10	10	01	1111111110111111
01	10	01	10	1111111111011111
01	10	01	01	1111111111101111
01	01	10	10	1111111111110111
01	01	10	01	1111111111111011
01	01	01	10	1111111111111101
01	01	01	01	1111111111111110

ISOPs with 8 PIs

x_1	x_2	x_3	x_4	X_5
01	01	01	01	01 $\cdots$ 15
10	11	11	11	0000000011111111
01	11	11	11	1111111100000000
11	10	11	11	0000111100001111
11	01	11	11	1111000011110000
11	11	10	11	0011001100110011
11	11	01	11	1100110011001100
11	11	11	10	0101010101010101
11	11	11	01	1010101010101010

Fig. 2. Positional cubes for two ISOPs of DEC_4.

The observations of Example 4.2 can be generalized as follows:

Theorem 4.2. *The function DEC_n has a WSOP that requires at least 2^n PIs and an MSOP that requires at most $2n$ PIs.*

The above theorem proves the existence of an n-variable 2^n-output variable function, where the sizes of the MSOP and the WSOP are at most $2n$ and at least 2^n, respectively. The upper ISOP for DEC_4 shown in Fig. 2 is not a WSOP since an ISOP with 20 PIs has been found for DEC_4.

5 Derivation of All ISOPs

Very little is known about the distribution of the sizes for ISOPs. For example, even for single-output functions, we know of no study that shows how many ISOPs exist with various number of product terms.

Although various methods to generate all the ISOPs for a logic function are known [22], [12], [21], [6], [35], [25], no experimental results have been reported. Experiments are computationally intensive even for functions with a small number of variables. However, we can obtain the statistical properties of ISOPs for some interesting functions.

Before showing the complete algorithm, consider the following:

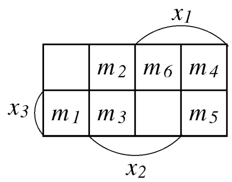


Fig. 3. $f(x_1, x_2, x_3) = ST(3,1)$.

Example 5.1.

$$f = ST(3,1) = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$$

has six minterms (Fig. 3) and six PIs

$$[p_1 : \bar{x}_1 x_3,\ p_2 : \bar{x}_1 x_2,\ p_3 : x_2 \bar{x}_3,\ p_4 : x_1 \bar{x}_3,\ p_5 : x_1 \bar{x}_2,\ p_6 : \bar{x}_2 x_3].$$

Fig. 4 is the covering table for $ST(3,1)$. It shows the following relations:

$$To\ cover\ m_1, p_1 \vee p_6\ is\ necessary.$$
$$To\ cover\ m_3, p_1 \vee p_2\ is\ necessary.$$
$$To\ cover\ m_2, p_2 \vee p_3\ is\ necessary.$$
$$To\ cover\ m_6, p_3 \vee p_4\ is\ necessary.$$
$$To\ cover\ m_4, p_4 \vee p_5\ is\ necessary.$$
$$To\ cover\ m_5, p_5 \vee p_6\ is\ necessary.$$

To satisfy all the conditions at the same time, we have $P(f) = 1$, where

$$P(f) = (p_1 \vee p_6)(p_1 \vee p_2)(p_2 \vee p_3)(p_3 \vee p_4)(p_4 \vee p_5)(p_5 \vee p_6).$$

$P(f)$ is called the Petrick function [22]. By expanding $P(f)$ into SOPs, we have

$$\begin{aligned} P(f) &= (p_1 \vee p_2 p_6)(p_3 \vee p_2 p_4)(p_5 \vee p_4 p_6) \\ &= p_1 p_3 p_5 \vee p_2 p_3 p_5 p_6 \vee p_1 p_2 p_4 p_5 \vee \underline{p_2 p_4 p_5 p_6} \\ &\quad \vee p_1 p_3 p_4 p_6 \vee p_2 p_3 p_4 p_6 \vee \underline{p_1 p_2 p_4 p_6} \vee p_2 p_4 p_6. \end{aligned}$$

Note that each product with an underline is covered by another product having fewer literals. Such products are redundant. Deleting these products, we have

$$\begin{aligned} P(f) = p_1 p_3 p_5 \vee p_2 p_3 p_5 p_6 \vee p_1 p_2 p_4 p_5 \vee p_1 p_3 p_4 p_6 \\ \vee p_2 p_3 p_4 p_6 \vee p_2 p_4 p_6. \end{aligned}$$

$P(f)$ consists of all the PIs of the Petrick function [23] and each PI of $P(f)$ corresponds to an ISOP for f. Furthermore, each literal p_i in the PI of $P(f)$ corresponds to a PI for f. For example, $p_1 p_3 p_5$ corresponds to the ISOP $\bar{x}_1 x_3 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_2$. Note that there are six ISOPs; two have three PIs, while four have four PIs. Thus, $ST(3,1)$ has two MSOPs with three PIs and four WSOPs with four PIs.

		minterms					
		m_1	m_3	m_2	m_6	m_4	m_5
	p_1	1	1				
	p_2		1	1			
	p_3			1	1		
PIs	p_4				1	1	
	p_5					1	1
	p_6	1					1

Fig. 4. Covering table of $ST(3,1)$.

In this way, all the ISOPs are obtained. For general functions, the number of minterms and PIs are very large. Thus, we use an ROBDD (reduced ordered binary decision diagram) to represent the function and a Prime_TDD (Ternary decision diagram) [31] to represent the set of all the PIs. In the Prime_TDD for f, each path from the root node to the constant 1 node corresponds to a PI for f. We also use an ROBDD to represent the Petrick function. While there are many ways to generate all the ISOPs of a given function f, we use the following algorithm:

Algorithm 5.1 (Generation of all ISOPs for a function f).

1. Generate all the PIs for f by using the Prime_TDD (the ternary decision diagram representing PIs) of f.
2. From the set of PIs and the set of minterms for f, generate the Petrick function $P(f)$ (which represents the covering table [22]).
3. Generate the Prime_TDD (which represents all the PIs) of $P(f)$.
4. Generate the 1-paths of the Prime_TDD and, for each 1-path, generate the corresponding ISOP.

In the Prime_TDD in Step 4, each path from the root node to the constant 1 corresponds to a PI for $P(f)$ and to an ISOP for f. Each 1 edge has weight 1 and each 0 edge has weight 0. The total sum of weights from the root node to the constant 1 nodes is the number of PIs in the ISOP. Note that the shortest path corresponds to an MSOP and the longest path corresponds to a WSOP.

6 Experimental Results

6.1 $ST(n,k)$ Functions

Using Algorithm 5.1, we compare the number of PIs in $ST(n,k)^r$ for different n, k, and r. Table 2 shows the number of PIs in the MSOP and the WSOP of $ST(n,k)^r$, as well as the total number of PIs. Shown also are the results of the Minato-Morreale algorithm.

The 9SYM (or SYM9) [11], [15] function shown in [3, p. 165] is identical to $ST(9,3)$. It has 1,680 PIs, $\tau(WSOP : ST(9,3)) \geq 148$, and $\tau(MSOP : ST(9,3)) = 84$. POP [9], a PRESTO-type [4], [33] logic minimization algorithm, produced an ISOP with 148 products. CAMP [1] produced an ISOP with 130 PIs, while MINI [15] did well, producing 85 PIs.

Table 3 shows the distribution of ISOPs to the number of PIs in an ISOP for $ST(n,1)$ for $3 \leq n \leq 7$. This data was obtained by Algorithm 5.1. It can be seen that the set of MSOPs is small compared to the set of *all* ISOPs.

6.2 Other Functions

We also applied Algorithm 5.1 to compare the number of PIs for multiple-output functions. Table 4 shows the distribution of the number of PIs in ISOPs for various arithmetic functions [32].

INCn is an n-input $n+1$ output function such that the value of the output is $x+1$, where x is the value of the

TABLE 2
Number of PIs and Redundancy Ratio for Various Functions

	n	MSOP	WSOP	ISOP MM	PI	ρ	σ
$ST(3,1)$	3	3	4	4	6	1.33333	1.1006
$ST(3,1)^2$	6	9	16	16	36	1.77778	1.1006
$ST(3,1)^3$	9	27	64	64	216	2.37037	1.1006
$ST(3,1)^4$	12	81	256	256	1296	3.16049	1.1006
$ST(3,1)^5$	15	243	1024	1024	7776	4.21399	1.1006
$ST(3,1)^6$	18	729	4096	4096	46656	5.61866	1.1006
$ST(4,1)$	4	4	6	6	12	1.50000	1.1067
$ST(4,1)^2$	8	16	36	36	144	2.25000	1.1067
$ST(4,1)^3$	12	64	216	216	1728	3.37500	1.1067
$ST(4,1)^4$	16	256	1296	1296	20736	5.06250	1.1067
$ST(4,1)^5$	20	1024	7776	7776	248832	7.59375	1.1067
$ST(5,1)$	5	5	8	8	20	1.60000	1.0986
$ST(5,1)^2$	10	25	64	64	400	2.56000	1.0986
$ST(5,1)^3$	15	125	512	512	8000	4.09600	1.0986
$ST(5,1)^4$	20	625	4096	4096	160000	6.55360	1.0986
$ST(5,2)$	5	10	14	14	30	1.40000	1.0696
$ST(6,1)$	6	6	10	10	30	1.66667	1.0889
$ST(6,2)$	6	15	24	24	90	1.60000	1.0815
$ST(7,1)$	7	7	12	12	42	1.71429	1.0800
$ST(7,2)$	7	21	†	36	210	†	†
$ST(7,3)$	7	35	†	50	420	†	†
$ST(8,1)$	8	8	14	14	56	1.75000	1.0725
$ST(8,2)$	8	28	†	50	420	†	†
$ST(9,1)$	9	9	16	16	72	1.77778	1.0660
$ST(9,2)$	9	36	†	66	756	†	†
$ST(10,1)$	10	10	18	18	90	1.80000	1.0605

n: number of input variables.
MSOP: number of PIs in MSOP.
WSOP: number of PIs in WSOP.
ISOP MM: number of PIs in an ISOP generated by Minato-Morreale's algorithm.
PI: number of prime implicants.
ρ: redundancy ratio: $\tau(WSOP:f)/\tau(MSOP:f)$
σ: normalized redundancy ratio: $\sqrt[n]{\rho}$
†: memory overflow precluded generation of these values

input; WGT5 is the same as RD53, a 5-input 3-output function, where the output is a binary number whose value is the number of 1s on the inputs; ROT6 computes the square root of a 6-bit integer; LOG5 computes the logarithm of the 5-bit integer; ADR3 is a 3-bit adder; and SQR5 computes the square of the 5-bit input.

Note that all the ISOPs for INC6 have the same number of PIs. This means any logic minimizer obtains an exact minimum solution. This is also is true for WGT5. For ADR3, most of the ISOPs have 31 PIs or 33 PIs. This is consistent with the observation that the logic minimization of ADR3 is relatively easy. For SQR5, the distribution is very wide. The MSOPs have 27 PIs, while WSOPs have 37 PIs. This is consistent with the observation that the minimization of SQR5 is more difficult. Note that SQR5 is a 10 output binary function. The data shown is for all outputs.

Although we could not obtain the distribution for SQR6 due to the memory overflow, we conjecture that the distribution of number of PIs for SQR6 is also wide. We also developed WIRR, a heuristic algorithm to obtain ISOPs with many products. For SQR5, SQR6, and 9SYM, the numbers of PIs in the solutions are shown in Table 5.

7 Distribution of ISOPs—An Analytic Approach

The distribution of ISOPs to the number of PIs is a way to represent the search space a heuristic algorithm must traverse in a minimization of an expression. For the case of $ST(n,1)$ functions, we can show a part of this distribution; a graph representation of the set of PIs allows this.

Definition 7.1. *Let F be an ISOP of $ST(n,1)$. In the* ***graph representation*** G_F ***of*** F

1. *G_F has nodes $x_1, x_2, \ldots,$ and x_n, and*
2. *G_F has an edge from x_i to x_j iff $\bar{x}_i x_j$ is a PI in F.*

TABLE 3
Distribution of ISOPs in $ST(n,1)$ Functions

# of PIs	$ST(3,1)$	$ST(4,1)$	$ST(5,1)$	$ST(6,1)$	$ST(7,1)$
3	2				
4	3	6			
5		36	24		
6		16	360	120	
7			560	3600	720
8			125	13530	37800
9				9270	282660
10				1296	435330
11					170352
12					16807
Total	5	58	1069	27816	943669

Example 7.1. Fig. 5 shows the graph representations of the MSOP and WSOP for $ST(3,1)$ (shown in Fig. 1).

We show that the graph representation of an ISOP of F has a special property.

Definition 7.2. *A directed graph G is* ***strongly connected*** *iff for every pair of vertices (a,b) in G, there is a path from a to b and from b to a. A directed graph G is* ***minimally strongly connected*** *iff it is strongly connected and the removal of any edge causes G not to be strongly connected.*

Theorem 7.1. *Let G_F be a graph representation of F. F is an ISOP of $ST(n,1)$ iff G_F is minimally strongly connected.*

Proof. See the Appendix. □

The graph representations of the MSOP and WSOP of $ST(3,1)$, shown in Fig. 5, are both strongly connected, as they should be by Theorem 7.1. Since each edge represents a prime implicant, an MSOP has a graph representation with the fewest edges. This observation facilitates the enumeration of MSOPs.

Theorem 7.2. *The number of MSOPs for $ST(n,1)$ is $(n-1)!$.*

Proof. See the Appendix. □

The graph representation allows a characterization of ISOPs. Specifically, complementing all variables in an ISOP of $ST(n,1)$ is equivalent to reversing the direction of all edges in the graph representation G_F of F. If G_F is minimally strongly connected, then the graph obtained from G_F by reversing the direction of all edges is also minimally strongly connected. This proves:

Lemma 7.1. *If F is an ISOP of $ST(n,1)$, then the SOP derived from F by complementing all variables is an ISOP of $ST(n,1)$.*

Example 7.2. When all variables are complemented, the graph representations of the ISOPs shown in Fig. 5 produce the graphs in Fig. 6, which also represent an MSOP and a WSOP.

It is important to note the difference between changing an ISOP F and changing the function realized by F. That is, an $ST(n,k)$ function is unchanged by a complementation of variables, i.e. it is a *self-anti-dual* function [32].

However, an ISOP for an $ST(n,k)$ function may or may not be changed when all variables are complemented. For example, $F = \bar{x}_1 x_2 \vee \bar{x}_2 x_3 \vee \bar{x}_3 x_1$ is an ISOP for $ST(3,1)$. Complementing all variables in F yields, $F = x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_1$, a different ISOP.

It is interesting that the WSOP for $ST(3,1)$ is unchanged by a complementation of all variables, as can be seen by comparing Fig. 5b with Fig. 6b. The invariance of an ISOP with respect to complementation of all literals is a unique characteristic of WSOPs, as shown in the next result.

Lemma 7.2. *Let F be an ISOP of $ST(n,1)$. F is a WSOP iff complementing all variables in F leaves F unchanged.*

Proof. See the Appendix. □

TABLE 4
Distribution of ISOPs in Arithmetic Functions

# of PIs	INC6	WGT5	ROT6	LOG5	ADR3	SQR5
22	13942125			12		
23			1088	286		
24			3072	1010		
25			2640	1326		
26			720	478		9452
27			64			281252
28						3599288
29						20725014
30						58836676
31		59049			447561	92597382
32					64224	83902808
33					449865	42813004
34					64224	11297310
35					2304	1340364
36						54176
37						280

TABLE 5
Number of PIs Produced by Various Algorithms on Three Benchmark Functions

Algorithm	SQR5	SQR6	$ST(9,3)$**
Quine-McCluskey [22]	26 MSOP	47 MSOP	84 MSOP
ESPRESSO-IIC [3]	26 MSOP	49	85
MINI-APL [15]	–	49	85
POP-C [9]	–	53	148
CAMP [1]	–	–	130
Minato-Morreale [19, 20]	30	58	148
WIRR*	31	71	136
Algorithm 5.1	37 WSOP	†	†

* Algorithm to find an ISOP with many PIs.
** $ST(9,3)$ is equivalent to 9SYM.
† Memory overflow precluded generation of these values.
– For these algorithms, there are no published results. We do not have the code.
Values marked MSOP and WSOP are known to be MSOP and WSOP.

It is interesting that Lemma 7.2 does not generalize to $ST(n,k)$. Specifically, for $ST(5,2)$, the ISOP

$$\begin{aligned} F = & \bar{x}_1\bar{x}_2x_3x_5 \vee \bar{x}_3\bar{x}_5x_1x_2 \vee \bar{x}_1\bar{x}_3x_2x_4 \vee \bar{x}_2\bar{x}_4x_1x_3 \\ & \vee \bar{x}_1\bar{x}_4x_2x_5 \vee \bar{x}_2\bar{x}_5x_1x_4 \vee \bar{x}_1\bar{x}_5x_3x_4 \vee \bar{x}_3\bar{x}_4x_1x_5 \\ & \vee \bar{x}_2\bar{x}_3x_4x_5 \vee \bar{x}_4\bar{x}_5x_2x_3 \end{aligned}$$

is invariant with respect to complementation of all variables. However, it is an MSOP and not a WSOP.

We can also enumerate WSOPs as follows:

Theorem 7.3. *The number of WSOPs for $ST(n,1)$ is n^{n-2}.*

Proof. See the Appendix. □

The graph representation allows the enumeration of other classes of ISOPs. For example, we can enumerate ISOPs that have one more PI than is in the MSOP. Specifically,

Theorem 7.4. *The number of ISOPs for $ST(n,1)$ with $n+1$ PIs is*

$$\frac{1}{2}\binom{n-1}{2}n!.$$

Proof. See the Appendix. □

By comparing the number of MSOPs with either the number of WSOPs or the number of ISOPs with one more PI than in the MSOP, we find that the former is much less than either of the latter for large n. That is, as n approaches infinity, the ratio of MSOPs to WSOPs approaches 0 (use Stirling's formula to replace $(n-1)!$ in the expression for the number of MSOPs). This proves the following:

Theorem 7.5. *The fraction of ISOPs for $ST(n,1)$ that are MSOPs approaches 0 as n approaches infinity.*

It is interesting that the ratio of the number of ISOPs with $n+1$ PIs (one more PI than is in an MSOP) to the number of WSOPs also approaches 0 as n approaches infinity. This suggests that WSOPs are much more common than minimal *or* near-minimal ISOPs.

8 Concluding Remarks

The existence of an algorithm that finds the worst sum-of-products expression for a class of functions is surprising. It counters our expectation that a heuristic algorithm should perform "reasonably" well. Also, the large difference between the size of the worst and the best expression is especially compelling since such an algorithm will perform very poorly. It is, therefore, an interesting question of whether there are other algorithms and other functions that exhibit the same characteristics.

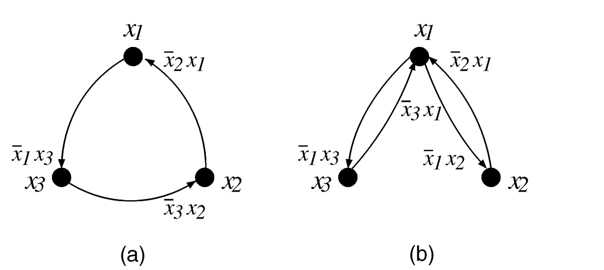


Fig. 5. Graph representations of the MSOP and WSOP for $ST(3,1)$. (a) MSOP. (b) WSOP.

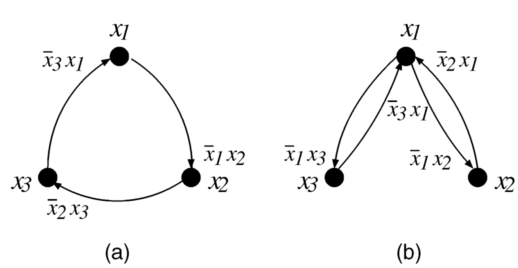


Fig. 6. Graph representations of Fig. 5 with all variables complemented. (a) MSOP. (b) WSOP.

We show a multiple-output function where the worst and the best ISOPs differ greatly in size. Specifically, a decoder with 2^n outputs and n inputs realizes a function where a WSOP has at least 2^n PIs and an MSOP has at most $2n$ PIs. Since this is a commonly used logic function, disparity in the size of WSOPs and MSOPs cannot be viewed as a characteristic of contrived functions only.

Although computationally intensive, enumeration of the ISOPs for representative functions gives needed insight into the problem. We show an algorithm to compute all ISOPs of a given function. We apply it to benchmark functions and show there are significant differences in the distributions of ISOPs. That is, some functions have a narrow distribution, where the WSOP is nearly or exactly the same size as the MSOP. These tend to be easy to minimize. For example, for unate functions [17] and parity functions, there is exactly one ISOP. Such functions are classified as "trivial" in the Berkeley PLA Benchmark Set (e.g., ALU1, BCD, DIV3, CLP1, CO14, MAX46, NEWPLA2, NEWBYTE, NEWTAG, and RYY6) [26]. Other functions display a wide range and tend to be hard to minimize. For example, 9SYM or SYM9 ($ST(9,3)$) has a wide range, i.e., the number of PIs in a WSOP and an MSOP is 148 and 84 PIs, respectively. This function is known to be hard to minimize.

For a class of functions, we provide an analysis showing that the number of MSOPs is significantly smaller than the number of WSOPs. That is, by showing a correlation with directed graphs, we enumerate all MSOPs and all WSOPs of the class and show that the number of MSOPs and WSOPs is $(n-1)!$ and n^{n-2}, respectively. As n increases, the ratio of PIs in a WSOP to the PIs in an MSOP grows without bound. This suggests such functions are hard to minimize.

A complete understanding of the minimization process will require knowledge of the search space and how various algorithms progress through it. However, such an understanding is not likely to be achieved in the near future. Our research suggests that there is merit to understanding the correlation between the degree of difficulty in minimizing a function and the distribution of its ISOPs.

Appendix

Lemma 2.1.

$$\tau(WSOP : S^7_{\{0,1,3,4,6,7\}}) = 70.$$

Proof. There are two steps. In the first step, we prove that an ISOP with 70 PIs exists for this function. In the second step, we show that it is a WSOP. For the first step, it is convenient to view the symmetric function as having three parts. Specifically,

$$S^7_{\{0,1,3,4,6,7\}} = S^7_{\{0,1\}} \vee S^7_{\{3,4\}} \vee S^7_{\{6,7\}}.$$

A WSOP is obtained by finding a WSOP of each of the three parts separately. Consider the 7-bit Hamming code shown in Table 6.

TABLE 6
A 7-Bit Hamming Code

Code word	Code word
0 0 0 0 0 0 0	1 1 1 1 1 1 1
0 0 0 1 1 1 1	1 1 1 0 0 0 0
0 0 1 0 0 1 1	1 1 0 1 1 0 0
0 0 1 1 1 0 0	1 1 0 0 0 1 1
0 1 0 0 1 0 1	1 0 1 1 0 1 0
0 1 0 1 0 1 0	1 0 1 0 1 0 1
0 1 1 0 1 1 0	1 0 0 1 0 0 1
0 1 1 1 0 0 1	1 0 0 0 1 1 0

For each code word, create a PI that covers two minterms by replacing *one* of the most abundant bits in the code word by a don't care. In the case of code word, 0000000, this creates seven PIs, each of which covers the minterm with all variables 0 and one minterm with exactly one 1. This covers all minterms of $S^7_{\{0,1\}}$. Similarly, seven PIs generated from code word 1111111 cover all minterms of $S^7_{\{6,7\}}$.

All of the remaining 14 code words have either four 0s and three 1s or four 1s and three 0s. For each, create four PIs by changing one of the four logic values in the majority to a don't care. Collectively, the four PIs cover the original code word and four words that are a distance one away from the code word. Because the distance between any pair of code words is at least three, a change in a single bit of a code word in the Hamming code creates a word that is not a code word that is distinct from either another code word or a word that is one bit different from another code word. This implies that those minterms a distance one away from a code word are covered by at most one PI. It follows that each PI is irredundant. Since each of the 14 code words corresponds to a set of PIs that cover five distinct minterms, there are $14 \times 5 = 70$ minterms total. On the other hand, the number of minterms for $S^7_{\{3,4\}}$ is $\binom{7}{3} + \binom{7}{3} = 35 + 35 = 70$. It follows that these PIs cover all the minterms of $S^7_{\{3,4\}}$. In all, $7 + 56 + 7 = 70$ PIs cover all of the $8 + 35 + 35 + 8 = 86$ minterms of the function. It follows that this set of PIs is a cover for $S^7_{\{0,1,3,4,6,7\}}$. Further, it is an irredundant cover and we have an ISOP.

We have proven that an ISOP with 70 PIs exists for $S^7_{\{0,1,3,4,6,7\}}$. We show that this is a WSOP by showing that no more than seven, 56, and seven PIs can cover the minterms in $S^7_{\{0,1\}}$, $S^7_{\{3,4\}}$, and $S^7_{\{6,7\}}$, respectively. Since $S^7_{\{0,1\}}$ and $S^7_{\{6,7\}}$ are monotone functions, their ISOPs are unique. Each consists of seven PIs. For $S^7_{\{3,4\}}$, the ISOP of 56 PIs above covers the 70 minterms associated with this function. On the contrary, assume that the proposed ISOP is not a WSOP. Thus, there is a set of $p > 56$ PIs that forms an ISOP for these minterms. Each PI covers exactly two minterms, for a total of $2p > 112$ instances of a PI covering a minterm. Let m_1 and $m_{>1}$ be the number of minterms covered by one and more than one PI, respectively. $m_{>1} = 70 - m_1$. Since the set of PIs is irredundant, each PI covers at least one minterm that is not covered by any other PI. Thus, $m_1 \geq p > 56$. It follows that $2p + 3m_1 > 280$. Further, $2p - m_1 > 280 - 4m_1 = 4(70 - m_1)$ and we can write

$$\frac{2p - m_1}{70 - m_1} > 4. \tag{A.1}$$

Here, the numerator is the number of instances in which a PI covers a minterm that is covered by more than one PI, while the denominator represents the number of minterms covered by more than one PI. Since this ratio exceeds four, by the Pigeonhole Principle, there is at least one minterm covered by at least five PIs. But, this is impossible; each minterm is covered by no more than four PIs (i.e., each code word is covered by a PI derived from a code word by converting one of the four most abundant variables, 0 or 1, to a don't care). Thus, it must be that the proposed ISOP is a WSOP. □

Theorem 3.1.

1. $\tau(CSOP : ST(n,k)) = \binom{n}{k, n-2k, k} = \frac{n!}{k!(n-2k)!k!}$.
2. $\tau(MSOP : ST(n,k)) = \binom{n}{k}$.
3. $\tau(WSOP : ST(n,k)) \geq 2\binom{n}{k} - \binom{2k}{k}$.

Proof.

1. An implicant of $ST(n,k)$ has the form

   $$x_{i_1} x_{i_2} \cdots x_{i_k} \bar{x}_{i_{n-k+1}} \bar{x}_{i_{n-k+2}} \cdots \bar{x}_{i_n}.$$

   That is, for this implicant to be 1, at least k variables ($x_{i_1}, x_{i_2}, \ldots$, and x_{i_k}) must be 1 and at least k variables must be 0 ($x_{i_{n-k+1}}, x_{i_{n-k+2}}, \ldots$, and x_{i_n}), where $2k \leq n$. This implicant is prime; deleting a literal creates an implicant that is 1 when $ST(n,k)$ should be 0. Specifically, deleting $x_{i_1}, x_{i_2}, \ldots$, or x_{i_k} creates an implicant that is 1 when less than n variables are 1, while deleting $\bar{x}_{i_{n-k+1}}, \bar{x}_{i_{n-k+2}}, \ldots$, or $\bar{x}_{i_n}$ creates an implicant that is 1 when more than $n-k$ variables are 1.

   The number of such PIs is the number of ways to separate n variables into three parts, where order within a part is not important. This is the multinomial $\binom{n}{k, n-2k, k} = \frac{n!}{k!(n-2k)!k!}$.

2. First, we show that $\binom{n}{k}$ is a lower bound on the number of PIs of $ST(n,k)$. Then, we show a set Π of $\binom{n}{k}$ PIs that covers all and only minterms in the function $ST(n,k)$. It follows that the OR of all PIs in Π is an MSOP for $ST(n,k)$.

   Consider the set MI of $\binom{n}{k}$ minterms of the form $\bar{x}_{i_1}\bar{x}_{i_2} \ldots \bar{x}_{i_k} x_{i_{k+1}} x_{i_{k+2}} \ldots x_{i_n}$, where $i_j \in \{1, 2, \ldots, n-1\}$ and $n \geq 2k$. That is, MI consists of minterms that are 1 when *exactly* k of the n variables are 0. No PI for $ST(n,k)$ covers two or more minterms in MI. As such, MI is a set of independent minterms and at least $\binom{n}{k}$ PIs are needed.

   Π is formed as follows: For each minterm mt in MI, apply Algorithm 1.1 below, producing P_{mt}, a PI that covers mt. Add P_{mt} to Π. Since no PI covers two or more minterms in MI, Π has $\binom{n}{k}$ distinct PIs. Since P_{mt} has exactly k 0s and k 1s, it covers *only* minterms in $ST(n,k)$. Next, we show that Π covers *all* minterms in $ST(n,k)$ by applying Algorithm 1.1 to an *arbitrary* minterm mt' of $ST(n,k)$, producing $P_{mt'}$, a PI that covers mt'. $P_{mt'} \in \Pi$, as follows: Form a minterm mt'' in MI from $P_{mt'}$ by setting all -s to 1s. Applying Algorithm 1.1 to mt'' yields $P_{mt''}$ that is identical to $P_{mt'}$, from which we can conclude $P_{mt'} \in \Pi$.

   Algorithm 1.1 (Produce a PI that covers a given minterm).
   Input: Minterm $mt = mt(0)mt(1)\ldots mt(n-1)$
   Output: Prime implicant

   $$P_{mt} = P_{mt}(0)P_{mt}(1)\ldots P_{mt}(n-1)$$

   (Initially, $P_{mt}(i) = -$ for all i where $0 \leq i \leq n-1$)

   1. $ZeroOnePairs = 0$
   2. Repeat until $ZeroOnePairs = k$ do {
      if

      $(P_{mt}(i)P_{mt}(i+1)\cdots P_{mt}(i+s) \in -\{0,1\}^{s-1}-$
      and $mt(i)mt(i+s) = 01)$

      then

      $\{P_{mt}(i)P_{mt}(i+s) \leftarrow 01$
      and $ZeroOnePairs \leftarrow ZeroOnePairs + 1\}$,

      where index addition is mod n (in this algorithm, we assume that subscript indices range from 0 to $n-1$, i.e., the variables are $x_0, x_1, \cdots, x_{n-1}$) and where $1 \leq s \leq n-1$.

   It is straightforward to show that Algorithm 1.1 produces a PI in Π if the minterm input has at least k 1s and k 0s. This PI may be different depending on the values of i chosen in each repetitive step.

3. Assume $\tau(WSOP : ST(n-1,k)) \geq 2\binom{n-1}{k} - \binom{2k}{k}$. We can form an ISOP of $ST(n,k)$ as follows:

   $$F = \bar{x}_n F_1 \vee x_n F_2 \vee F_3,$$

   where

   a. F_1 is an SOP such that each product term is formed as the AND of i) a set X_1 of $k-1$ complemented variables, where $X_1 \subseteq X - \{x_n\}$, and of ii) k uncomplemented variables from $X - \{x_n\} - X_1$, where the indices of the uncomplemented variables are all as small as possible (given the choice of X_1). Because X_1 can be chosen in $\binom{n-1}{k-1}$ ways, F_1 has $\binom{n-1}{k-1}$ product terms.

   b. F_2 is an SOP such that each product term is formed as the AND of i) a set X_2 of $k-1$ uncomplemented variables, where $X_2 \subseteq X - \{x_n\}$ and of ii) k complemented variables from $X - \{x_n\} - X_2$, where the indices of the complemented variables are all as small as possible. Because X_2 can be chosen in $\binom{n-1}{k-1}$ ways, F_2 has $\binom{n-1}{k-1}$ product terms.

c. F_3 is one of the WSOPs for $ST(n-1,k)$. From the inductive hypothesis, F_3 has $2\binom{n-1}{k} - \binom{2k}{k}$ PIs.

F is an expression for $ST(n,k)$ as follows: Consider a minterm m in $ST(n,k)$. If m has at least k 1s and at least k 0s, *regardless* of the value of x_n, then m is covered by F_3. If m has exactly k 0s, including a 0 value for x_n, and at least k 1s, then it is covered by $\bar{x}_n F_1$. If m has exactly k 1s, including a 1 value for x_n, and at least k 0s, it is covered by $x_n F_2$. Thus, F covers all minterms in $ST(n,k)$. Because each PI in $\bar{x}_n F_1$ and $x_n F_2$ (and also in F_3) has exactly k uncomplemented and exactly k complemented variables, F covers only minterms with at least k 0s and at least k 1s, i.e., only minterms in $ST(n,k)$. It follows that F is an SOP for $ST(n,k)$.

If F has no redundant PIs, its $\binom{n-1}{k-1} + \binom{n-1}{k-1} + 2\binom{n-1}{k} - \binom{2k}{k}$ or $2\binom{n}{k} - \binom{2k}{k}$ PIs form an ISOP for $ST(n,k)$. Thus,

$$\tau(WSOP : ST(n,k)) \geq 2\binom{n}{k} - \binom{2k}{k}.$$

Next, we show that F has no redundant PIs. First, each PI in $\bar{x}_n F_1$ covers a minterm m' having k 0s and $n-k$ 1s that is not covered by any combination of PIs from $x_n F_2$ and F_3. Thus, no PI in $\bar{x}_n F_1$ is redundant. By a similar argument, no PI in $x_n F_2$ is redundant. Second, no PI in F_3 is redundant, as follows: Since F_3 is a WSOP for $ST(n-1,k)$, no PI in F_3 is covered by the OR of one or more PIs in F_3. If the OR of PIs from $\bar{x}_n F_1$ and $x_n F_2$ covers a PI P from F_3, then it follows that at least one product term P_1 in F_1, when ANDed with at least one product term P_2 in F_2, yields a non-0 result. Let

$$P_1 = \bar{x}_{s_1}\bar{x}_{s_2}\ldots\bar{x}_{s_{k-1}}x_{t_1}x_{t_2}\ldots x_{t_k}$$

and

$$P_2 = x_{u_1}x_{u_2}\ldots x_{u_{k-1}}\bar{x}_{v_1}\bar{x}_{v_2}\ldots\bar{x}_{v_k}.$$

If $P_1P_2 \neq 0$, no s_i is the same as a u_j and no t_p is the same as a v_q. But, $t_1, t_2, \ldots,$ and t_k were chosen to be as small as possible without overlapping $s_1, s_2, \ldots,$ and s_{k-1}, while $v_1, v_2, \ldots,$ and v_k were chosen to be as small as possible without overlapping $u_1, u_2, \ldots,$ and u_{k-1}. Consider the indices $I = \{1, 2, \ldots, k\}$. The smallest index in I that appears neither in $S = \{s_1, s_2, \ldots, s_{k-1}\}$ nor in $U = \{u_1, u_2, \ldots, u_{k-1}\}$ appears in both $T = \{t_1, t_2, \ldots, t_k\}$ and $V = \{v_1, v_2, \ldots, v_k\}$, causing $P_1P_2 = 0$, a contradiction. □

Theorem 3.5. *$ST(m,k)^r$ has the following properties:*

1. $\tau(CSOP : ST(m,k)^r) = \binom{m}{k,\, m-2k,\, k}^r = (\frac{m!}{k!(m-2k)!k!})^r$.
2. $\tau(MSOP : ST(m,k)^r) = \binom{m}{k}^r$.
3. $\tau(WSOP : ST(m,k)^r) \geq [2\binom{m}{k} - \binom{2k}{k}]^r$.

Proof. Items 1, 2, and 3 follow from the observation that a minterm in the product function $ST(m,k)^r$ can be viewed as the AND of a minterm from each of the factor functions $ST(m,k)$. Thus, a PI of $ST(m,k)^r$ can be viewed as the product of a PI from each $ST(m,k)$ and Item 1 follows directly.

Also,

$$\tau(MSOP : ST(m,k)^r) \leq \binom{m}{k}^r$$

follows directly. That is, $\binom{m}{k}^r$ is an upper bound on the number of PIs in an MSOP for $ST(m,k)^r$, as an ISOP for $ST(m,k)^r$ can be formed as the AND of PIs from the MSOPs of $ST(m,k)$. As is shown by Voight and Wegener [36], certain product functions can have fewer PIs in their MSOPs than the product of the number of PIs in the MSOPs of the factor functions. However, when the factor functions are $ST(m,k)$, we can observe the following: Let M be the set of minterms covered by $ST(m,k)$, in which exactly k variables are uncomplemented. Since the PIs of $ST(m,k)$ have exactly k uncomplemented and k complemented variables, none cover two or more minterms in M. It follows that at least $\binom{m}{k}$ PIs are needed to cover $ST(m,k)$. It follows that at least $\binom{m}{k}^r$ PIs are needed to cover the minterms in $ST(m,k)^r$ that are the product of minterms in the set M for each factor function. Thus,

$$\tau(MSOP : ST(m,k)^r) = \binom{m}{k}^r.$$

A WSOP for $ST(m,k)^r$ can be formed as the product of WSOPs for each factor function. Since $2\binom{m}{k} - \binom{2k}{k}$ is a lower bound on the number of PIs in each factor function, Item 3 follows directly. □

Theorem 7.1. *Let G_F be the graph representation of F. F is an ISOP of $ST(n,1)$ iff G_F is minimally strongly connected.*

Proof. (if) Consider a minimally strongly connected digraph G_F, where F is its corresponding SOP. Thus, for every edge (x_j, x_i) in G_F, there is an implicant $\bar{x}_j x_i$ in F. On the contrary, assume that F is not an ISOP for $ST(n,1)$. That is, either 1) F does not cover $ST(n,1)$ or 2) F covers $ST(n,1)$ but has a redundant PI.

Consider 1). If F does not cover $ST(n,1)$, then there is a minterm mt such that mt either a) has no complemented variables or no uncomplemented variables, but F is 1 for this assignment or b) has at least one uncomplemented variable and at least one uncomplemented variable, but F is 0 for this assignment. The first part, a), is not possible; all PIs cover only minterms with at least one complemented variable and at least one uncomplemented variable. The second part, b), is also not possible, as follows: Because G_F is strongly connected, there is a path $x_j = x_{k_1}, x_{k_2}, \cdots, x_{k_m} = x_i$, from x_j to x_i. Since $x_j = 0$ and $x_i = 1$, the assignment of values to $x_{k_1}, x_{k_2}, \ldots,$ and x_{k_m} corresponding to mt has the property that there is an s such that $x_{k_s} = 0$ and $x_{k_s+1} = 1$. The corresponding PI $\bar{x}_{k_s} x_{k_s+1}$ is in F and is 1 for the assignment associated with mt. Thus, F covers mt.

Consider 2). If F covers $ST(n,1)$, but has a redundant PI, $\bar{x}_j x_i$, then F', which is F with $\bar{x}_j x_i$ removed, also covers $ST(n,1)$. But, $G_{F'}$ is G_F with one edge, $(x_j x_i)$, removed. It follows that G_F is not *minimally* strongly connected.

(only if) Let G_F be a graph representation of an ISOP F of $ST(n,1)$. Assume, on the contrary, that G_F is not minimally strongly connected. That is, either 1) G_F is not strongly connected or 2) G_F is strongly connected, but is not minimal.

If G_F is not strongly connected, there are two nodes x_j and x_i such that no path exists from x_j to x_i. Let $Suc(x_j)$ be the set of all nodes for which there is a path *from* x_j, i.e., all *successors* of x_j. Let $Pre(x_i)$ be the set of all nodes for which there exists a path *to* x_i, i.e., all *predecessors* of x_i. Consider a minterm mt that is 0 for x_j and all variables associated with nodes in $Suc(x_j)$ and is 1 for x_i and all variables associated with nodes in $Pre(x_i)$. Since there is no path from x_j to x_i, $Suc(x_j) \cap Pre(x_i) = \phi$ and such an assignment assigns exactly one value to nodes in $Suc(x_j) \cup Pre(x_i) \cup \{x_j, x_i\}$. Choose the values of all other variables to be 1 (or 0). No edge in G_F has a 0 at its tail and a 1 at its head. Thus, all PIs are 0 and F is not an SOP for $ST(n,1)$. It is, thus, not an ISOP, a contradiction.

If G_F is strongly connected, but is not minimal, there is at least one edge (x_j, x_i) that can be removed without affecting the connectedness of G_F. It follows that $G_{F'}$, where F' is F with $\bar{x}_j x_i$ removed, is a graph representation of F', an SOP for the same function as F. Thus, F is an SOP, but not an ISOP, contradicting the assumption. □

Theorem 7.2. *The number of MSOPs for* $ST(n,1)$ *is* $(n-1)!$.

Proof. From Theorem 7.1, an MSOP for $ST(n,1)$ corresponds to a directed graph with the fewest edges which is strongly connected, but is not strongly connected when any edge is removed. Such a graph is a directed cycle of arcs through all variables. As such, it represents a cyclic permutation on the variables. The number of such permutations is $(n-1)!$. □

Lemma 7.2. *Let* F *be an ISOP of* $ST(n,1)$. F *is WSOP iff complementing all variables in* F *leaves* F *unchanged.*

Proof. (if) Let F be an ISOP that is unchanged by a complementation of all variables. This implies that if $\bar{x}_i x_j$ is a PI of F, then so also is $\bar{x}_j x_i$. It follows that if (x_i, x_j) is an edge in G_F, then (x_j, x_i) is an edge in G_F. That is, all edges between nodes occur in pairs, one going one way and the other going the other way. In such a graph, there are $n-1$ pairs or $2n-2$ edges in all. (Replace each pair by an undirected edge. If the directed graph is strongly connected, the undirected graph must be connected. From Harary [14], there must be $n-1$ edges.) Since there are $2n-2$ edges in G_F, there are $2n-2$ PIs in F and, thus, F is a WSOP.

(only if) Let F be a WSOP of $ST(n,1)$. We show that G_F consists of cycles of length 2 only. Thus, if $\bar{x}_i x_j$ is a PI of F, so also is $\bar{x}_j x_i$. It follows that complementing all variables of F leaves F unchanged. Suppose that G_F contains a cycle of length m, where $m > 2$. Such a cycle represents a strongly connected subgraph of G_F in which there are m edges. However, the cycle can be replaced by a minimally strongly connected graph with more edges (e.g., where all edges occur in pairs). The result is a strongly connected graph, where the deletion of an edge leaves it unconnected, which has more edges than the original version. This contradicts the statement that F is a WSOP. □

Theorem 7.3. *The number of WSOPs for* $ST(n,1)$ *is* n^{n-2}.

Proof. From Theorem 7.1, a WSOP for $ST(n,1)$ corresponds to a minimally strongly connected graph with the largest number of edges. We show that this graph consists of cycles of length 2 exclusively, as follows: Suppose, on the contrary, the graph has a cycle of length $m > 2$. There are m edges in this cycle. However, this subgraph can be replaced by a subgraph with more edges, $2(m-1)$. It follows that the original graph does not represent a WSOP.

Each cycle of length 2 connects two nodes by edges in the two directions. Replace each pair of edges by an undirected edge, forming an undirected tree with $n-1$ edges. Thus, there are $2(n-1)$ PIs in a WSOP for $ST(n,1)$. This proves Theorem 3.2. It follows that the number of WSOPs is the number of undirected trees on n labeled nodes. Cayley [5] in 1889 showed that this number is n^{n-2}. □

Theorem 7.4. *The number of ISOPs for* $ST(n,1)$ *with* $n+1$ *PIs is*

$$\frac{1}{2}\binom{n-1}{2} n!.$$

Proof. From Theorem 7.1, an ISOP of $ST(n,1)$ that has $n+1$ PIs corresponds to a minimally strongly connected graph with $n+1$ edges. All graphs with this property have two cycles of nodes, of which i are common, where $1 \le i \le n-2$. We can represent each instance as a permutation of nodes that has been divided into three nonempty sets, the i common nodes, nodes N_1, in one cycle only, and nodes N_2 in the other cycle only. N_1 and N_2 must be nonempty since an empty set corresponds to a redundant edge. There are $n!$ ways to permute the n nodes and $\binom{n-1}{2}$ ways to divide them into three nonempty sets. However, this double counts graphs since interchanging N_1, and N_2 does not change the graph. Thus, the total number of graphs with $n+1$ edges is $\frac{1}{2}\binom{n-1}{2} n!$. □

ACKNOWLEDGMENTS

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science, and Technology of Japan and in part by NPS Direct Funded Grant UHCD1. Discussions with Dr. Shin-ichi Minato were quite useful. Mr. Munehiro Matsuura and Mr. Shigeyuki Fukuzawa did part of the experiments. The authors acknowledge the contributions of two referees. This paper is an extended version of T. Sasao and J.T. Butler, "Comparison of the Worst and Best Sum-of-Products Expressions for Multiple-Valued Functions," *Proceedings of the IEEE International Symposium on Multiple-Valued Logic,* pp. 55-60, May 1997.

REFERENCES

[1] N.N. Biswas, "Computer-Aided Minimization Procedure for Boolean Functions," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 5, no. 2, pp. 303-304, Apr. 1986.

[2] A. Blake, "Canonical Expressions in Boolean Algebra," dissertation, Dept. of Math., Univ. of Chicago, 1937.

[3] R.K. Brayton, G.D. Hachtel, C.T. McMullen, and A.L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis.* Boston: Kluwer Academic, 1984.

[4] D.W. Brown, "A State-Machine Synthesizer—SMS," *Proc. 18th Design Automation Conf.,* June l981.

[5] A. Cayley, "A Theorem on Trees," *Quarterly J. Math.,* vol. 23, pp. 376-378, 1889.

[6] D.M.Y. Chang and T.H. Mott Jr., "Computing Irredundant Normal Forms Abbreviated Presence Functions," *IEEE Trans. Electronic Computers,* vol. 14, pp. 335-342, June 1965.

[7] O. Coudert, "Two-Level Logic Minimization: An Overview," *Integrated VLSI J.,* vol. 17, pp. 97-140, Oct. 1994.

[8] M. Dagenais, V. Agrawal, and N. Rumin, "McBOOLE: A New Procedure for Exact Logic Minimization," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 5, no. 1, pp. 229-223, Jan. 1986.

[9] G. De , M. Hofmann, R. Newton, and A. Sangiovanni-Vincentelli, "A Design System for the PLA-Based Digital Circuits," *Advances in Computer-Aided Eng. Design,* vol. 1, pp. 285-364, Jay Press, 1985.

[10] D.L. Dietmeyer, *Logic Design of Digital Systems,* second ed. Boston: Allyn and Bacon, 1978.

[11] B. Dunham and R. Fridshal, "The Problem of Simplifying Logical Expressions," *J. Symbolic Logic,* vol. 24, pp. 17-19, 1959.

[12] M.J. Ghazala, "Irredundant Disjunctive and Conjunctive Forms of a Boolean Function," *IBM J. Research and Development,* pp. 171-176, Apr. 1957.

[13] M.A. Harrison, *Introduction to Switching and Automata Theory.* McGraw-Hill, 1965.

[14] F. Harary, *Graph Theory.* Addison-Wesley, 1969.

[15] S.J. Hong, R.G. Cain, and D.L. Ostapko, "MINI: A Heuristic Approach for Logic Minimization," *IBM J. Research and Development,* pp. 443-458, Sept. 1974.

[16] W.H. Kautz, "A Survey and Assessment of Progress in Switching Theory and Logical Design in the Soviet Union," *IEEE Trans. Electronic Computers,* vol. 15, no. 2, pp. 164-204, Apr. 1966.

[17] R. McNaughton, "Unate Truth Functions," *IRE Trans. Electronic Computers,* vol. 10, no. 1, pp. 1-6, 1961.

[18] A.R. Meo, "On the Synthesis of Many-Variable Switching Functions," *Networks and Switching Theory,* G. Biorci, ed., chapter VI, pp. 470-482, New York: Academic Press, 1968.

[19] S. Minato, "Fast Generation of Prime-Irredundant Covers from Binary Decision Diagrams," *IEICE Trans. Fundamentals,* vol. E76-A, no. 6, pp. 976-973, June 1993.

[20] E. Morreale, "Recursive Operators for Prime Implicant and Irredundant Normal Form Determination," *IEEE Trans. Computers,* vol. 19, no. 6, pp. 504-509, June 1970.

[21] T.H. Mott Jr., "Determination of Irredundant Normal Forms of a Truth Function by Iterated Consensus of the Prime Implicants," *IEEE Trans. Electronic Computers,* vol. 9, pp. 245-252, June 1960.

[22] S. Muroga, *Logic Design and Switching Theory.* Wiley-Interscience, 1979.

[23] R.J. Nelson, "Simplest Normal Truth Functions," *J. Symbolic Logic,* pp. 105-108, June 1954.

[24] K. Nguyen, M. Perkowski, and N. Goldstein, "Palmini-Fats Boolean Minimizer for Personal Computers," *Proc. Design Automation Conf.,* pp. 615-621, Aug. 1987.

[25] B. Reusch, "Generation of Prime Implicants from Subfunctions and a Unifying Approach to the Covering Problem," *IEEE Trans. Computers,* pp. 924-930, Sept. 1975.

[26] R. Rudell and A. Sangiovanni-Vincentelli, "Multiple-Valued Minimization for PLA Optimization," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 6, no. 5, pp. 727-750, Sept. 1987.

[27] T. Sasao, "An Application of Multiple-Valued Logic to a Design of Programmable Logic Arrays," *Proc. 18th Int'l Symp. Multiple-Valued Logic,* pp. 65-72, May 1978.

[28] T. Sasao, "Multiple-Valued Decomposition of Generalized Boolean Functions and the Complexity of Programmable Logic Arrays," *IEEE Trans. Computers,* vol. 30, no. 9, pp. 635-643, Sept. 1981.

[29] T. Sasao, "Input Variable Assignment and Output Phase Optimization of PLA's," *IEEE Trans. Computers,* vol. 33, no. 10, pp. 879-894, Oct. 1984.

[30] *Representation of Discrete Functions,* T. Sasao and M. Fujita, eds. Kluwer Academic, 1996.

[31] T. Sasao, "Ternary Decision Diagrams and Their Applications," *Representation of Discrete Functions,* T. Sasao and M. Fujita, eds., chapter 12, Kluwer Academic, 1996.

[32] T. Sasao, *Switching Theory for Logic Synthesis,* pp. 269-270. Kluwer Academic, 1999..

[33] A. Svoboda and D.E. White, *Advanced Logical Circuit Design Techniques.* New York: Garland Press, 1979.

[34] Texas Instruments, *The TTL Data Book for Design Engineers,* pp. 308-311.Texas Instruments, 1973.

[35] P. Tison, "Generalization of Consensus Theory and Application to the Minimization of Boolean Functions," *IEEE Trans. Electronic Computers,* pp. 446-456, Aug. 1967.

[36] B. Voight and I. Wegener, "A Remark on Minimal Polynomials of Boolean Functions," *CSL '88, Second Workshop Computer Science Logic Proc.,* pp. 372-383, 1989.

[37] S.V. Yablonski, "The Problem of Bounding the Length of Reduced Disjunctive Normal forms," *Prob. Kiber.,* vol. 7, pp. 229-230, in Russian, 1962.

Tsutomu Sasao received the BE, ME, and PhD degrees in electronics engineering from Osaka University, Osaka, Japan, in 1972, 1974, and 1977, respectively. He has held faculty/research positions at Osaka University, Japan, the IBM T.J. Watson Research Center, Yorktown Heights, New York, and the Naval Postgraduate School, Monterey, California. He is now a professor of in the Department of Computer Science and Electronics, as well as the director of the Center for Microelectronic Systems at the Kyushu Institute of Technology, Iizuka, Japan. His research areas include logic design and switching theory, representations of logic functions, and multiple-valued logic. He has published more than eight books on logic design, including *Logic Synthesis and Optimization*, *Representation of Discrete Functions*, and *Switching Theory for Logic Synthesis* (Kluwer Academic, 1993, 1996, 1999, respectively). He has served as program chairman for the IEEE International Symposium on Multiple-Valued Logic (ISMVL) many times. Also, he was the symposium chairman of the 28th ISMVL held in Fukuoka, Japan, in 1998. He received the NIWA Memorial Award in 1979 and Distinctive Contribution Awards from the IEEE Computer Society MVL-TC in 1987 and 1996 for papers presented at ISMVLs. He has served as an associate editor of the *IEEE Transactions on Computers*. He is a fellow of the IEEE.

Jon T. Butler received the BEE and MEngr degrees from Rensselaer Polytechnic Institute, Troy, New York, in 1966 and 1967. He received the PhD degree from the Ohio State University, Columbus, in 1973. Since 1987, he has been a professor at the Naval Postgraduate School, Monterey, California. From 1974 to 1987, he was at Northwestern University, Evanston, Illinois. During that time, he served two periods of leave at the Naval Postgraduate School, first as a National Research Council Senior Postdoctoral Associate (1980-1981) and second as the NAVALEX Chair Professor (1985-1987). He served one period of leave as a foreign visiting professor at the Kyushu Institute of Technology, Iizuka, Japan. His research interests include logic optimization and multiple-valued logic. He has served on the editorial boards of the *IEEE Transactions on Computers*, *Computer*, and the IEEE Computer Society Press. He has served as the editor-in-chief of *Computer* and IEEE Computer Society Press. He has received the Award of Excellence, the Outstanding Contributed Paper Award, and a Distinctive Contributed Paper Award for papers presented at the International Symposium on Multiple-Valued Logic. He has received the Distinguished Service Award, two Meritorious Service Awards, and nine Certificates of Appreciation for service to the IEEE Computer Society. He is a fellow of the IEEE.

▹ **For further information or this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.